aBc
INFORMATIQUE

Page manquante
(colophon)

aBC INFORMATIQUE

VOLUME 6

EDITIONS ATLAS

Éditions Atlen, Bruxelles
Finabuch s.a., Éditions Transalpines, Mezzovico
Éditions Atlas Canada Ltée, Montréal Nord

Sommaire

Le marché

Un bras mécanique peut se comporter « intelligemment ». Nous allons voir ce qu'il faut faire pour construire un robot qui fasse réellement preuve d'« intelligence » en se déplaçant 1201

Après avoir rendu le robot « intelligent » par une programmation adéquate, examinons maintenant s'il peut « voir » les objets qui l'entourent 1221

Il est difficile de doter un robot de la parole. Pour comprendre les problèmes liés au langage, il est nécessaire d'examiner les théories qui décrivent l'apprentissage de la parole chez l'homme 1241

Nous avons déjà expliqué de façon détaillée les différents « sens » qui contribuent à créer l'« intelligence » d'un robot. Examinons ici comment ces « sens » peuvent être combinés 1261

Nous avons étudiés les méthodes à mettre en œuvre pour que les robots se comportent intelligemment. A présent, regardons comment simuler le comportement d'un robot par ordinateur 1281

Nous allons examiner certains des produits mis en vente sous le nom de robots. Nous verrons s'ils remplissent bien le rôle d'un robot et s'ils satisfont à la définition très précise que nous en avons donnée . 1301

Parmi les bras de robot haut de gamme disponibles sur le marché, deux catégories se dégagent : les robots utilisés à des fins didactiques et ceux qui représentent le nec plus ultra de la technique 1321

Avec le développement des marchés, les éditeurs de logiciels poussent à l'adaptation et à la traduction des logiciels en différentes langues. De gros intérêts sont en jeu. Attention aux pièges 1361

La communication entre ordinateurs signifie que des données sont transmises des uns aux autres. Comment s'établissent ces transferts qui passent le plus souvent par les lignes du téléphone? 1381

Comment fonctionne un système de communication mettant en œuvre un micro-ordinateur et un modem? Qu'est-ce qu'un « duplex »? C'est ce que nous nous proposons de vous expliquer 1401

Nous allons voir les principes généraux que le programmeur doit observer dans la conception de programmes destinés à des enfants des écoles maternelles et primaires 1421

Boîte à outils

Il est possible de produire des signaux sonores à partir d'un convertisseur numérique/analogique. Nous allons étudier la production de divers signaux et déterminer la durée d'une note 1212

Nous avons déjà conçu un code machine servant à générer trois types d'ondes sonores : carrées, en dents de scie et sinusoïdales. Examinons deux autres paramètres : volume et hauteur 1226

Nous avons déjà mis au point le matériel et le logiciel servant à piloter un véhicule à deux moteurs. Voyons un programme devant piloter ce véhicule à l'intérieur d'un labyrinthe 1252

Nous avons mis au point un système électronique à tampon qui peut être utilisé avec le Commodore 64 pour contrôler et commander des périphériques extérieurs 1278

Nous allons entreprendre la réalisation d'un nouveau projet : construire un robot d'appartement équipé de détecteurs de lumière et de palpeurs. Nous nous bornerons ici à présenter ce projet dans ses grandes lignes et à donner quelques détails sur l'assemblage de la carrosserie et du moteur 1290

Notre robot utilise des moteurs électriques appelés moteurs « pas à pas ». Faisant usage de signaux logiques, ils sont donc adaptés à un contrôle de type numérique 1315

Après l'installation des moteurs et la construction du circuit, nous allons connecter les moteurs à la fiche D et mettre au point une interface très simple avec le port utilisateur. Nous rédigerons aussi un programme permettant de tester le robot 1336

La première phase du montage de notre robot est terminée, et nous l'avons testée en écrivant un court programme. Nous allons à présent monter quatre capteurs à micro-interrupteurs sur le robot et écrire un programme simple pour tester leur fonctionnement 1356

Après les moteurs et les capteurs, nous allons étalonner le robot pour nous permettre de commander de façon précise les distances et les angles de ses déplacements 1375

En connectant un servomoteur au port utilisateur de votre ordinateur et en utilisant le système tampon que nous avons déjà mis au point, il est possible d'utiliser diverses commandes 1392

Nous avons vu comment connecter un servomoteur à un port utilisateur via le système tampon mis au point précédemment. Voici comment commander plusieurs servomoteurs simultanément 1403

Nous revenons à notre projet de robot pour concevoir un programme qui lui permettra de localiser et de mesurer précisément un côté d'un objet rectangulaire 1423

Logiciel

Notre série d'articles sur les tableurs et modèles financiers a été jusqu'ici consacrée au progiciel Vu-Calc de Psion. Nous étudions ici Abacus, un tableur également de chez Psion destiné au QL Sinclair 1204

Malgré de nombreuses ressemblances, les logiciels de modèles financiers gardent chacun leurs particularités. C'est le cas de Graph Plan, qui combine tableur et logiciel graphique 1232

Les programmes de simulation de vol sont devenus chose courante, mais Flyerfox, de Tymac, a sur eux un avantage évident : il parle, et ce, sans avoir besoin de périphérique supplémentaire 1240

Nous nous sommes déjà intéressés aux tableurs. Examinons maintenant Multiplan, un modèle particulièrement sophistiqué destiné au Commodore 64 et dû à Microsoft 1244

Mugsy est un jeu de stratégie produit par Melbourne House. Ici, le joueur prend la place de Mugsy, chef de bande menant un racket de « protecteur », dans l'Amérique des années trente 1260

Nous poursuivons notre série d'articles sur les tableurs par l'étude des macro-instructions sur Lotus 1-2-3, tableur avec base de données incorporée et fonctions graphiques 1264

Nos articles sur les tableurs se poursuivent par une étude sur TK! Solver, programme de gestion de modèles mis au point par les créateurs de VisiCalc . 1284

Nous terminons notre exposé sur TK! Solver — programme de traitement d'équations pour l'Apple II, l'IBM PC et ses compatibles, l'Apricot d'ACT..... 1304

« Logiciel vertical » est un mot du jargon informatique donné aux programmes spécialisés écrits pour des catégories socio-professionnelles bien déterminées 1324

Les Jeux Olympiques de Los Angeles ont donné des idées aux firmes productrices de logiciels. C'est le cas, en particulier, d'une entreprise américaine qui propose l'attrayant Summer Games 1340

Le programme BrainStorm de Caxton Software a été utilisé par une association de parents de jeunes héroïnomanes pour organiser des campagnes de sensibilisation du public contre les drogues dures ... 1344

Nous allons examiner quatre programmes basés sur des tableurs : Micro Swift, Practicalc II, PS et Vizastar. Ces produits confirment que les micros personnels ont de quoi tenir tête aux micros de gestion . 1366

Lors de l'apparition des premiers micro-ordinateurs, la grande question était de savoir ce que les individus pourraient en faire. Est-ce vraiment très différent aujourd'hui ? 1384

Quand l'ordinateur se fait précieux auxiliaire dans l'éducation des enfants, LOGO est un des langages privilégiés. Cherchons à savoir le pourquoi et le comment 1406

Examinons un ensemble de programmes qui s'appuient sur l'évaluation psychologique des personnes pour aider les décideurs à adopter une stratégie dans leurs rapports professionnels 1426

Jeux

Ce jeu de Mick Wilson vous propose une bataille navale contre votre ZX81. Il suit les règles du jeu habituel. L'extension mémoire de 16 K peut être utile. . . 1208

Voici un jeu d'adresse plus difficile qu'il n'y paraît. Son auteur, Pierre Monsaut, l'a conçu pour le micro-ordinateur Alice, le premier de la génération de chez Matra 1237

Vous voici seul, abandonné sur une planète défendue par des robots meurtriers. Le sol est truffé de mines. C'est « Robots », écrit par Pierre Monsaut pour le Commodore 64 1249

Ce jeu de Roland Mariott nécessite 24 K de mémoire. Il vous permet de jouer contre l'ordinateur Atari ou contre un adversaire de votre choix 1274

Sur une planète défendue par des robots meurtriers, vous êtes seul, et le sol est truffé de mines... Vous connaissez le problème. Mais peut-être pas avec le MO5 de Thomson 1296

Voici la deuxième partie d'un jeu dont nous avons déjà donné un programme pour votre ordinateur Atari. N'oubliez pas que les manettes de jeu sont obligatoires 1309

Écrit par Pierre Monsaut pour le micro-ordinateur Alice de chez Matra, ce jeu agréable est aussi un bon exercice de programmation pour d'éventuelles créations de jeux 1332

Ce jeu a été écrit par Paul Dunning pour le micro-ordinateur Atari. Le manche à balai est indispensable pour poser votre vaisseau spatial sur la lune . 1368

Le succès de ce jeu, plus connu sous l'expression « rocher, ciseaux et papier », ne peut être ignoré par votre ordinateur. Voici une version pour l'Oric, écrite en BASIC par Peter Shaw 1388

A petits pas rapides, Pierre Monsaut a écrit ce jeu en BASIC pour l'ordinateur domestique Vic 20. Un conseil : enregistrez-le avant de le faire tourner .. 1428

Matériel

L'achat d'un ordinateur domestique n'est que la première étape de la réalisation d'un système complet avant d'adapter les périphériques offerts sur le marché 1209

Philips se lance dans la micro-informatique avec un micro-ordinateur construit par sa filiale française RTC, destiné surtout aux jeunes et aux petits budgets : le VG 5000 1229

La robotique est devenue un élément essentiel de l'industrie informatique. Mais, jusqu'à présent, les robots ont surtout été utilisés dans des applications de l'industrie lourde. Avec Beasty, le robot se domestique 1250

CBM vient de lancer deux micro-ordinateurs destinés à l'usage personnel et à la petite gestion : le Plus/4

et le Commodore 16, moins onéreux mais dont la mémoire est plus réduite 1269

Le Sinclair Spectrum a connu un grand succès. Mais des nouveaux venus ont semblé surclasser cette vénérable machine. Son constructeur a donc décidé de lui offrir une livrée toute neuve 1286

Bien des périphériques ont été inventés pour faciliter la création graphique. Le double avantage de la table graphique Touchmaster est qu'elle fonctionne avec la plupart des ordinateurs les plus répandus 1310

La compagnie américaine Osborne Corp. fut la première à lancer un micro vraiment portable, l'Osborne-1. Après bien des ennuis, elle sort l'Osborne Encore, lui aussi compatible avec l'IBM PC 1329

Jusqu'ici, nous avons passé en revue tous les aspects du comportement d'un robot. Pour conclure, voyons quelles sont les limites des modèles actuels, et quels progrès on peut attendre 1341

Les micro compatibles IBM se multiplient. Ils sont généralement moins chers et plus performants que la machine qu'ils imitent. Voici un des premiers portables compatibles IBM, le Compaq Plus 1349

L'Alice 90, le nouvel ordinateur familial de Matra, a su conserver la simplicité d'accès à la micro-informatique d'Alice, tout en disposant de possibilités beaucoup plus larges 1369

Les derniers appareils photographiques exploitent la puissance des microprocesseurs pour faciliter les prises de vue. Nous passons en revue certains de ces appareils qui sont offerts sur le marché 1389

Nous avons déjà examiné de près la gamme d'ordinateurs de poche Casio. Jetons maintenant un coup d'œil sur les deux machines Sharp, assez différentes l'une de l'autre 1409

L'absence d'interfaces intégrées de lecteurs de disquettes avait compromis le succès de la série de micros Memotech 500. Memotech vient donc d'introduire le RS128 disposant de plusieurs interfaces. . 1429

Programmation

Comment modifier notre programme de recherche de variable de telle façon qu'il puisse modifier le nom de telle ou telle variable? Nous donnons ici deux versions 1206

Pour mettre en œuvre sur le Spectrum notre programme de remplacement de variable, il ne faut pas placer notre utilitaire dans une autre zone mémoire que le programme qu'il modifie, mais le fusionner à la fin de celui-ci 1224

Les jeux d'aventures sont très appréciés des possesseurs d'ordinateurs. Mais, s'il est passionnant d'y jouer, il est plus intéressant d'en écrire soi-même. Voici comment faire 1246

Les deux jeux d'aventures que nous créons sont de type « textuel » : le programme affiche à l'écran la description du lieu. Mettons au point un utilitaire pour formater ces messages 1272

Dans notre projet de programmation d'un jeu d'aventures, écrivons des programmes qui établissent des positions à l'intérieur du jeu et permettent aux joueurs de se déplacer 1293

Dans notre projet d'aventures, nous regardons ici comment le programme analyse et obéit aux instructions qui lui sont données par le joueur 1306

Nous avons déjà mis au point des routines permettant au joueur de prendre des objets. Voyons maintenant de quelle façon il peut les abandonner. Nous nous intéresserons aussi aux « lieux spéciaux » . . 1346

Nous allons compléter le programme de notre jeu d'aventures avec un sous-programme chargé de « générer » au hasard des fantômes qui hanteront la forêt 1363

Nous allons voir comment créer des écrans graphiques représentant l'unité arithmétique et logique (ALU) et le port manche à balai du micro-ordinateur ZX Spectrum 1412

Nous avons déjà, pour notre jeu d'aventures Digitaya, créé des écrans graphiques illustrant des lieux traversés par le joueur. Voici une troisième version, destinée au Commodore 64 1432

? Mots de passe

LOGO n'est pas un langage de premier choix pour les applications supposant beaucoup de calculs, mais il offre un tableau impressionnant de primitives de calcul numérique 1215

Le traitement de liste représente une notion centrale pour la compréhension de LOGO. En ajoutant une pincée de récursion, nous allons écrire un peu de poésie avec LOGO 1234

Le traitement de liste permet à LOGO de s'appliquer aux jeux. Nous montrons ici comment utiliser ce langage pour le développement d'un jeu d'aventures fondé sur le texte 1254

Nous avons vu comment définir avec LOGO les procédures de base d'un jeu d'aventures. Nous traitons ici des procédures de déplacement d'objets graphiques entre les pièces 1266

Notre projet de créer un jeu d'aventures avec LOGO arrive ici à sa fin. Après avoir défini les diverses positions et écrit les procédures pour se déplacer de l'une à l'autre, attardons-nous sur les détails 1288

L'utilisation des caractéristiques de traitement de liste de LOGO se poursuit par la constitution d'une base de données simple. Nous avons pris comme exemple une enquête policière sur un meurtre ... 1312

Dans ce dernier article sur la programmation LOGO, nous montrons comment ajouter de nouvelles structures de commandes au langage et des procédures capables d'écrire elles-mêmes d'autres procédures . 1333

Après le développement de routines et de procédures de gestion des données, ainsi que la création de dessins par la tortue, nous allons utiliser LOGO pour afficher les données de manière visuelle 1353

Nous commençons une courte série d'articles sur l'utilisation de LOGO pour la création de motifs géométriques. Nous allons tracer des cycloïdes, qui sont des courbes obtenues à partir de cercles 1372

LOGO est un langage très utile pour étudier la représentation de formes dans l'espace. Pour effectuer des transformations de formes, étudions une procédure qui modifie d'autres procédures 1395

Nous avons déjà étudié divers motifs symétriques. Aujourd'hui, nous abordons les figures à deux dimensions. Voyons comment définir avec LOGO les grilles ou treillis qui sont à la base de ces figures 1414

Nous allons voir comment LOGO est utilisé pour le calcul factoriel, et par quels moyens certains résultats sont transformés en « arborescence factorielle » ... 1434

Langage machine

Pour écrire des programmes plus longs et plus ambitieux, nous allons voir de nouvelles techniques qui nous permettront de structurer des programmes en langage d'assemblage 1218

Pour illustrer les techniques de conception « haut en bas » en langage d'assemblage, voici un programme de « débogage ». La première chose à faire est de développer le module de contrôle 1238

Complétons le développement de notre programme de « débogage » par l'ensemble des routines dont nous aurons besoin pour le module qui manie les entrées et les sorties 1257

Dans cette suite de notre programme de « débogage », nous allons compléter le module de routines pour manier les points d'interruption 1275

Nous allons considérer le mécanisme d'interruption utilisé pour transférer le contrôle entre le programme de débogage et le programme à déboguer aux points d'interruption 1297

Notre cours de langage d'assemblage se termine. Nous réunissons tous les morceaux de notre programme de débogage, codons le module principal proprement dit. 1318

Avec le Commodore 64, on peut translater facilement l'emplacement des données d'écran en mémoire. Voyons une routine pour concevoir et stocker plusieurs (jusqu'à huit) affichages écran 1398

Beaucoup de jeux de types arcades utilisent un défilement d'arrière-plan pour donner l'impression de mouvement rapide. Voici une routine pour faire défiler l'arrière-plan horizontalement le long de l'écran du C64 1417

Les pionniers

Les satellites de télécommunications permettent de faire communiquer les ordinateurs d'un continent à l'autre. Avec le Télécom 1 français, la télématique s'impose réellement 1358

Le premier micro-ordinateur du monde était français. L'activité de Micral est aujourd'hui reprise par le groupe Bull. Celui-ci manifeste sa volonté de s'imposer sur le marché micro-informatique 1438

Organigramme

On n'est pas toujours obligé d'utiliser ce type d'organigramme, mais, dans les cas les plus complexes, il met en évidence la logique du programme de manière très claire 1352

Nous réunissons ici des dessins géométriques qui nous servirons ultérieurement de symboles dans les organigrammes de flux, à mesure que ceux-ci deviendront plus complexes 1378

Pour avoir la possibilité de répéter une série d'opérations autant de fois que nous le souhaitons, il nous faut introduire la notion de « boucle », très utile en programmation 1386

S'il fallait trouver un équivalent concret d'une boucle en informatique, on pourrait songer à une noria. Mais dans l'un et l'autre cas, il faut savoir arrêter le mouvement 1408

Les « boucles à l'intérieur d'autres boucles » ou « boucles emboîtées » sont utiles, pratiques et de conception élégante, mais délicates à traiter 1437

Base de données

Voici, avec l'aimable autorisation de Zilog Inc., une autre partie de la carte référence du programmeur Z80. A ne pas manquer pour tous ceux qui veulent programmer 1220

Voici reproduit, avec l'aimable autorisation de Zilog Inc., la fin de la carte référence qui vient en complément du programme Z80, et de nos articles sur le langage machine 1300

Livres

Un fil d'Ariane. L'ordinateur à la portée de tous. — Initiation à l'informatique. — Aimeriez-vous comprendre l'informatique? — L'informatique, premier contact 1380

Introduction à l'informatique — Vous avez dit Micro? — Comprendre l'informatique — Papa, maman, l'ordinateur et moi 1387

L'ordinateur et l'informatique en 15 leçons — La micro-informatique et son abc — Micro-ordinateurs : comment ça marche? — Principes des ordinateurs 1420

Ainsi naquit l'informatique — Votre ordinateur et vous — Votre premier ordinateur 1440

Errata

Volume 5

P. 973, colonne 1, dans le listage, remplacer la ligne 2120 par : **2120 IF score$ < le_plus grand_score$ ...** Même chose pour la ligne 2110 de la colonne 2 ; remplacer la ligne 2190 (colonne 1) par : **2190 PRINT :PRINT rouge$; « Temps »** ; ... Colonne 2, ajouter à la fin de la ligne 2200 : $; remplacer dans tout ce listage COULEUR par **COLOR**.

Dans l'encadré jaune, remplacer la ligne 1410 par : **1410 UNTIL Y > 700** ; et la ligne 1480 par : **1480 UNTIL difficulté > − 1 AND difficulté 10.**

P. 974 remplacer la ligne 2080 par : **2080 UNTIL TEMPS > 12099 OR_DRAPEAU FIN = 1** et la ligne 2120 par : **2120 IF score$ le_plus_grand_score$ THEN le plus grand score$** ; remplacez la ligne 2160 par : **2160 PRINT rouge$...**, la ligne 2180 par : **2180 PRINT : PRINT rouge$; « le_plus_grand_score » ; TAB(30) ; le_plus_grand_score$** ; la ligne 2210 par : **2210 autre_partie$ =** ... la ligne 2330 par : **2330 xdet = 2:ydet = 25:** ... la ligne 2800 par : **2800 PRINT TAB(2,29) « Le plus grand score »** ; remplacez les lignes 3310 à 3340 par : **3310 IF xdet > 17 ..., 3320 IF ydet > 25 ..., 3330 IF xdet < 2 ..., 3340 IF ydet < 1 ...** ;

remplacez la ligne 3400 par : **3400 PROCconversion(xdet,ydet)** ; la ligne 3520 par : **3520 UNTIL x >xfin** ... et la ligne 3900 par : **3900 IF total > 4** et remplacer dans tout le listage SON par **SOUND**.

P. 978, au listage « Chargeur Remplisub », remplacer le début de la ligne 40 par : **40 READ A: IF CC<>A THEN** ...

P. 985, listage « BBC Micro », supprimer la virgule à la fin de la ligne 1020. Dans le listage Spectrum, écrire **GOTO** et **GOSUB** en un mot.

P. 995, colonne 2, remplacer les lignes 30, 150 et 550 par : **30 POKE DDR, 0: REM ENTRÉE SEULEMENT ; 150 PRINT «DATREG = «;PE;» = »;B$; 550 B$ = « »:N = PE.**

P. 996 : supprimer les lignes 170 et 180 qui sont écrites deux fois. A la ligne 190, ajouter : avant REM. A la fin de la ligne 750 lire : **... PRINT CH;«OK»:RETURN.**

P. 1008, encadré jaune. Dans « Compte d'impulsion (i) », colonne 4, remplacer la ligne 70 par : **70 IF (TI-T) > 3600 THEN 50** ; dans « Compte d'impulsions (ii) », colonne 2, remplacer la ligne 90 par : **90 UNTIL TEMPS>6000** ; colonne 4, remplacer les lignes 70 et 80 par **70 IF (TI—T)>3600 THEN 40** et **80 FOR K = 0 TO 7** ; dans « Combinaison », colonne 1, remplacer la fin de la ligne 100 par : **... ELSE PRINT « MAUVAISE COMBINAISON »** ; colonne 2, remplacer la ligne 55 par : **55 PRINT « ENTREZ UN CHIFFRE »**.

P. 1013, encadré jaune, ligne 4, écrire **VIDEECRAN** en un mot.

P. 1022, colonne 2, ligne 14, lire : **32 × 160**.

P. 1025, encadré vert, ligne 9, lire : **EDIT « CARRÉ** (sans guillemets après le nom) ; remplacer partout GH par **GA** (GAuche), CL par **LP** et CB par **PP**.

P. 1044, mêmes remarques, et remplacer AR par **RE** (REcule).

P. 1053, colonne 2, remplacer la ligne 4140 par : **4140 PE = PE + (TG*2 − 1)* (2^(8 − CP))**, et la ligne 61300 par : **61300 POKE 10, PEEK(10) OR 4**, et le début de la ligne 61400 par : **61400 POKE PO**, ...

P. 1060, légende à droite, lire : $\overline{\mathbf{M}}$ complément de M.

P. 1066, colonne 2, Commodore 64, ligne 90, remplacer AS par **A$**.

P. 1067, encadré rouge, listage 1, à la ligne 20, remplacer ! par : A la fin de la ligne 70, lire **<> 192** ; à la ligne 100 ; remplacer le premier < par **(**.

P. 1068, colonne 2, ligne 4, lire : **entre &0C00 et &0CFF**. A la fin de la ligne 5 et début ligne 6 : **entre 224 et 255**.

P. 1069, colonne 1, aux lignes 1460 et 1470, remplacer 1 par **i** :

FOR i = 1 TO 4
READ o(k,i):NEXT i

Ligne 1820, écrire **GOSUB** en un mot ; colonne 3, pour les noms de procédures, remplacer − par _. Écrire la variable **COMMANDE** en majuscules ; compléter la ligne 4580 comme suit (colonne 4) : **4580 UNTIL CHNOM>223 AND CHNOM<256.**

P. 1073, colonne 2, ligne 39, lire : **GA 30 RE :GRAND.**

P. 1074, colonne 2, ligne 2, remplacer 1,05 par **1.05**; dans l'encadré, remplacer TRACER par **DESSINE**.

P. 1075, dans les listages « LOGO », remplacer partout GH par **GA**, Ar par **RE**, CL par **LP** et CB par **PP**.

P. 1078, figure en bas, à gauche, remplacer $0017 par **$0517**.

P. 1103, dans le second listage, remplacer GH 30 par **GA 30**.

P. 1116, encadré bleu, listage Commodore 64 à la ligne 2010 remplacer − : par **− 1** ; ligne 2060, compléter par : **2060 IF (TI − T)<DR(I,2) THEN 2060**, et ligne 2030 du listage BBC : **2030 IF DR(I,1) = A(J) THEN ?** ...

P. 1127, encadré bleu-gris, listage 2, compléter la ligne 60 : **60 IF (TI − T)<300 THEN 60**.

P. 1128, dans les deux listages, compléter les lignes 460 : **460 IF CH<> − 32 THEN** ...

P. 1133 et 1134, remplacer toutes les commandes à l'infinitif (FAIRE, AFFICHER, etc.) par l'impératif (**FAIS, AFFICHE**, etc.). Remplacer partout GH par **GA** et AR par **RE**.

P. 1140, remplacer la légende de la figure par : « **Carte Apple-Tell**. Insérée dans l'une des fentes de l'Apple II, elle le transforme en Minitel intelligent. »

P. 1145, écrire GOTO en un mot ; remplacer le début de la ligne 9110 par : **9110 LET ligneno =** ... A la ligne 30210, écrire **Nouvelleligne** (en un mot).

P. 1146 à 1148, remplacer AR (arrière) par **RE** (recule), CL par **LP**, CB par **PP** et tous les verbes à l'infinitif par des impératifs.

P. 1155, aux lignes 280 et 290, remplacer le chiffre 0 par la lettre **O** ; à la fin de la ligne 380 ajouter une parenthèse fermante.

P. 1167, compléter la ligne 110 : **110 IF CC<255 THEN 60** ; dans le tableau, en haut à droite, à la ligne correspondant à 3, colonne D0, remplacer 0 par **1**.

P. 1174 à 1176, remplacer toutes les commandes LOGO à l'infinitif par des impératifs ; même chose pages 1186 et 1188.

P. 1194, colonne 2, listage BBC, remplacer value par **VALEUR**.